Лекция 19

16 апреля

Сложение двух чисел

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .text
global CMAIN
```

```
CMAIN:
   finit
   fld dword [x]
   fld dword [y]
   faddp
   fstp dword [z]
   PRINT_HEX 4, z
   NEWLINE
   xor eax, eax
   ret
```

x

11.2 ~ 1.11999998092651367187 5E1

0x41333333 = 01000001 00110011 00110011 00110011

Сложение двух чисел

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .text
global CMAIN
```

```
CMAIN:
   finit
   fld dword [x]
   fld dword [y]
   faddp
   fstp dword [z]
   PRINT_HEX 4, z
   NEWLINE
   xor eax, eax
   ret
```

y

0.7 ~ 6.999999880790710449218 75E-1

0x3F333333 = 00111111 00110011 00110011 00110011

Сложение двух чисел

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .text
global CMAIN
```

```
CMAIN:
   finit
   fld dword [x]
   fld dword [y]
   faddp
   fstp dword [z]
   PRINT_HEX 4, z
   NEWLINE
   xor eax, eax
   ret
```

z

0x413E6666 = 01000001 00111110 01100110 01100110

1.18999996185302734375E1 ~ 11.9

Печать числа

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .rodata
 lc db '%f', 10, 0

section .text
CEXTERN printf
global CMAIN
```

```
CMAIN:
    ; …
    ; пролог функции
    sub     esp, 20

    fld     dword [x]
    fld     dword [y]
    faddp
    fst     dword [z]
    fstp    dword [esp + 4]
    mov     dword [esp], lc
    call    printf

    add     esp, 20
    ; эпилог функции
    ; …
```

Печать числа

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .rodata
 lc db '%f', 10, 0

section .text
CEXTERN printf
global CMAIN
```

printf печатает мусор
где ошибка?!?!!1

```
CMAIN:
    ; …
    ; пролог функции
    sub     esp, 20

    fld     dword [x]
    fld     dword [y]
    faddp
    fst     dword [z]
    fstp    dword [esp + 4]
    mov     dword [esp], lc
    call    printf

    add     esp, 20
    ; эпилог функции
    ; …
```

Печать числа

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .rodata
 lc db '%f', 10, 0

section .text
CEXTERN printf
global CMAIN
```

ISO/IEC 9899:1999
§ 6.5.2.2 абзац №6

```
CMAIN:
    ; …
    ; пролог функции
    sub     esp, 20

    fld     dword [x]
    fld     dword [y]
    faddp
    fst     dword [z]
    fstp    qword [esp + 4]
    mov     dword [esp], lc
    call    printf

    add     esp, 20
    ; эпилог функции
    ; …
```

Польская обратная запись

```
(w + x + y + z) / 4
```

```
w x + y + z + 4 /
```

```
section .data
 w dq  1e10
 x dq  1e10
 y dq  1e10
 z dq  1e10
 d dd  4
```

```
section .text
    ;…
    fld     qword [w]
    fld     qword [x]
    faddp
    fld     qword [y]
    faddp
    fld     qword [z]
    faddp
    fild    dword [d]
    fdivp ; st1 / st0
    ;…
```

Порядок действий имеет значение

```
%include 'io.inc'

section .data
 x dq  3.14
 y dq  1e50
 z dq -1e50

section .bss
 r resq 1

section .rodata
 lc db '%f', 10, 0

section .text
CEXTERN printf
global CMAIN
```

```
CMAIN:
   ; … пролог функции
   sub     esp, 20

   fld     qword [x]
   fld     qword [y]
   fld     qword [z]
   faddp
   faddp
   fst     qword [r]
   fstp    qword [esp + 4]
   mov     dword [esp], lc
   call    printf

   add     esp, 20
   ; … эпилог функции
```

Порядок действий имеет значение

```
CMAIN:
    ; … пролог функции
    sub     esp, 20

    fld     qword [y]
    fld     qword [z]
    fld     qword [x]
    faddp
    faddp
    fst     qword [r]
    fstp    qword [esp + 4]
    mov     dword [esp], lc
    call    printf

    add     esp, 20
    ; … эпилог функции
```

```
CMAIN:
    ; … пролог функции
    sub     esp, 20

    fld     qword [x]
    fld     qword [y]
    fld     qword [z]
    faddp
    faddp
    fst     qword [r]
    fstp    qword [esp + 4]
    mov     dword [esp], lc
    call    printf

    add     esp, 20
    ; … эпилог функции
```

Распределение слагаемых

```
section .data
 x dq  1e200
 y dq  1e200
 z dq  1e200

section .bss
 r resq 1

section .rodata
 lc db '%lf', 10, 0
```

```
CMAIN:
   ; … пролог функции
   sub     esp, 20

   fld     qword [x]
   fld     qword [y]
   fsubp
   fld     qword [z]
   fmulp
   fst     qword [r]
   fstp    qword [esp + 4]
   mov     dword [esp], lc
   call    printf

   add     esp, 20
   ; … эпилог функции
```

Распределение слагаемых

```
section .data
 x dq  1e200
 y dq  1e200
 z dq  1e200

section .bss
 r resq 1

section .rodata
 lc db '%lf', 10, 0
```

```
CMAIN:
   ; … пролог функции
   sub     esp, 20

   fld     qword [x]
   fld     qword [z]
   fmulp
   fld     qword [y]
   fld     qword [z]
   fmulp
   fsubp
   ; вызов printf

   add     esp, 20
   ; … эпилог функции
```

Распределение слагаемых

```
section .data
 x dq  1e200
 y dq  1e200
 z dq  1e200

section .bss
 r resq 1
 cw resw 1

section .rodata
 lc db '%lf', 10, 0

CMAIN:
   ; … пролог функции
   sub     esp, 20
```

```
   fstcw   word [cw]
   and  word [cw], 11111111_11000000b
   fldcw   word [cw]

   fld     qword [x]
   fld     qword [z]
   fmulp
   fld     qword [y]
   fld     qword [z]
   fmulp
   fsubp
   ; вызов printf

   add     esp, 20
   ; … эпилог функции
```

Предопределенные константы

- На «верхушку» стека регистров (`St0`) помещается определенная константа
  - `FLD1` +1.0
  - `FLDL2T` $\log_2 10$
  - `L2E` $\log_2 e$
  - `FLDPI` π
  - `FLDLG2` $\log_{10} 2$
  - `FLDLN2` $\log_e 2$
  - `FLDZ` +0.0

Сравнение чисел

```
_Bool isLe(double x, float y) {
   return x <= y;
}
```

Результат сравнения	C3	C2	C0
St0 > St1	0	0	0
St0 < St1	0	0	1
St0 == St1	1	0	0
неопределенно	1	1	1

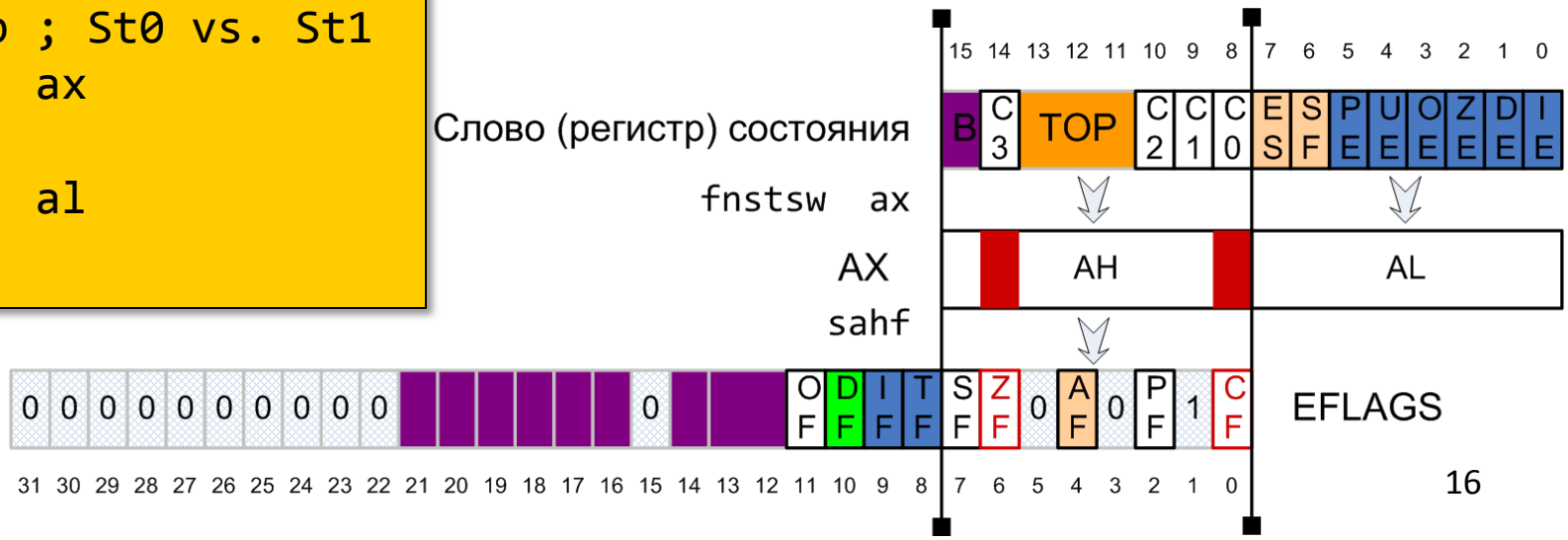
```
isLe:
    push    ebp
    mov     ebp, esp
    fld     dword [ebp+16]
    fld     qword [ebp+8]
    fucompp ; St0 vs. St1
    fnstsw  ax
    sahf
    setbe   al
    pop     ebp
    ret
```

Извлечение результатов сравнения

- C3 → ZF, C0 → CF
- Можно использовать условные коды, применяемые при сравнении беззнаковых чисел

Результат сравнения	ZF	PF	CF
St0 > St(*i*)	0	0	0
St0 < St(*i*)	0	0	1
St0 == St(*i*)	1	0	0
неупорядочены	1	1	1

```
isLe:
    ...
    fld     dword [ebp+16]
    fld     qword [ebp+8]
    fucompp ; St0 vs. St1
    fnstsw  ax
    sahf
    setbe   al
    ...
```



Непосредственное обновление EFLAGS

- F[U]COMI[P] St(*i*)
  - появились в P6

isLe(2.0, 3.0)

```
isLe:
   ...
   fld     dword [ebp+16]
   fld     qword [ebp+8]
   fucomip ; st0 vs. st1
   setbe   al
   ...
```

BE:
CF = 1 или ZF = 1

Результат сравнения	ZF	PF	CF
St0 > St(*i*)	0	0	0
St0 < St(*i*)	0	0	1
St0 == St(*i*)	1	0	0
неупорядочены	1	1	1

Функции: возвращаемое значение – число с плавающей точкой

```
void caller(double *p) {
   *p = inverse(*p);
}
```

```
float inverse(double x) {
   return 1/x;
}
```

```
caller:
   push    ebp
   mov     ebp, esp
   sub     esp, 8
   mov     eax, dword [ebp+8]
   fld     dword [eax]
   fstp    qword [esp]
   call    inverse
   mov     eax, dword [ebp+8]
   fstp    qword [eax]
   leave
   ret
```

На входе в функцию регистры St0 – St7 должны быть пустыми

```
inverse:
   push    ebp
   mov     ebp, esp
   fld1
   fld     qword [ebp+8]
   fdivp
   pop ebp
   ret
```

На выходе из функции:

1. регистры St1 – St7 должны быть пустыми,
2. регистр St0 либо содержит возвращаемое значение, либо см. п. №1

Некоторые команды x87

- FABS
- FCHS
- FSQRT
- FXCH [ST(*i*)]
- FTST

- FYL2X
- F2XM1

- FCOS
- FSINCOS
- FPTAN
- FPATAN

- FILD m[16/32/64]int
- FIST[P] m[16/32/64]int

- FICOM[P] m[16/32/64]int

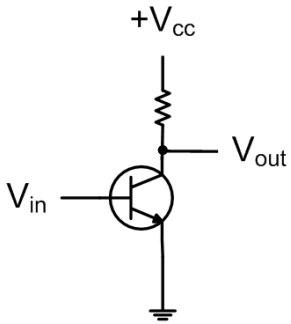
Итоги второй части

- Процессор с точки зрения пользовательской программы
- Основные механизмы реализации языка Си
  - Соответствие типов
  - Указатели, адресная арифметика
  - Управляющие операторы: разные стратегии реализации
  - Размещение данных в памяти
    - Классы памяти: статическая, автоматическая, динамическая
    - Выравнивание
  - Соглашения о вызове функции
- ABI – Application Binary Interface
- IA-32: x86 + x87 + ...
  - Некоторые особенности устройства

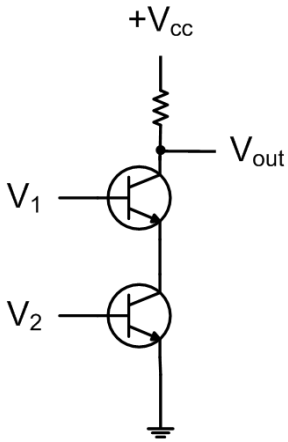
Далее: Архитектура ЭВМ

- Организация аппаратного обеспечения компьютера
  - Физические основы, шины, периферийные устройства
- Организация памяти
  - НЖМД, кэш
  - Производительность
- Организация ЦПУ
  - Конвейер, система команд
- Многозадачная работа компьютера
  - Изоляция пользовательских программ

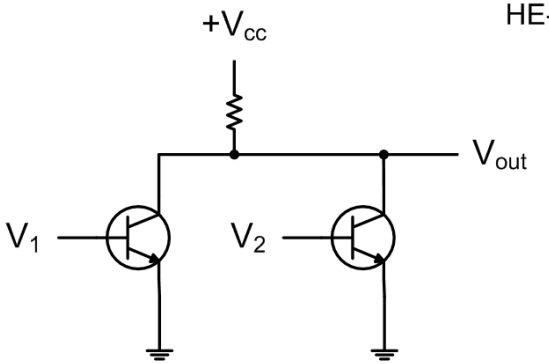
Логические вентили



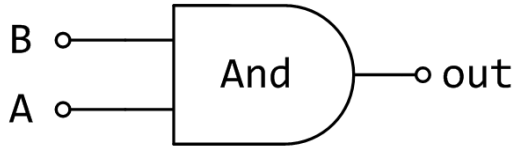
Инвертор

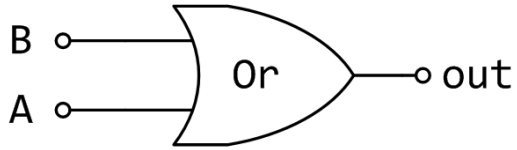


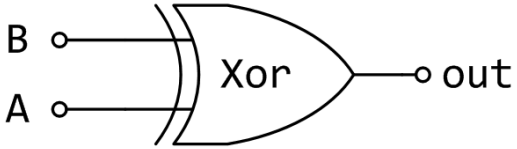
НЕ-И

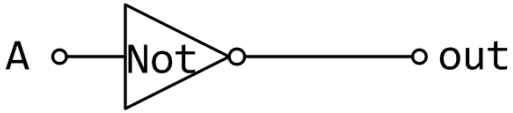


НЕ-ИЛИ

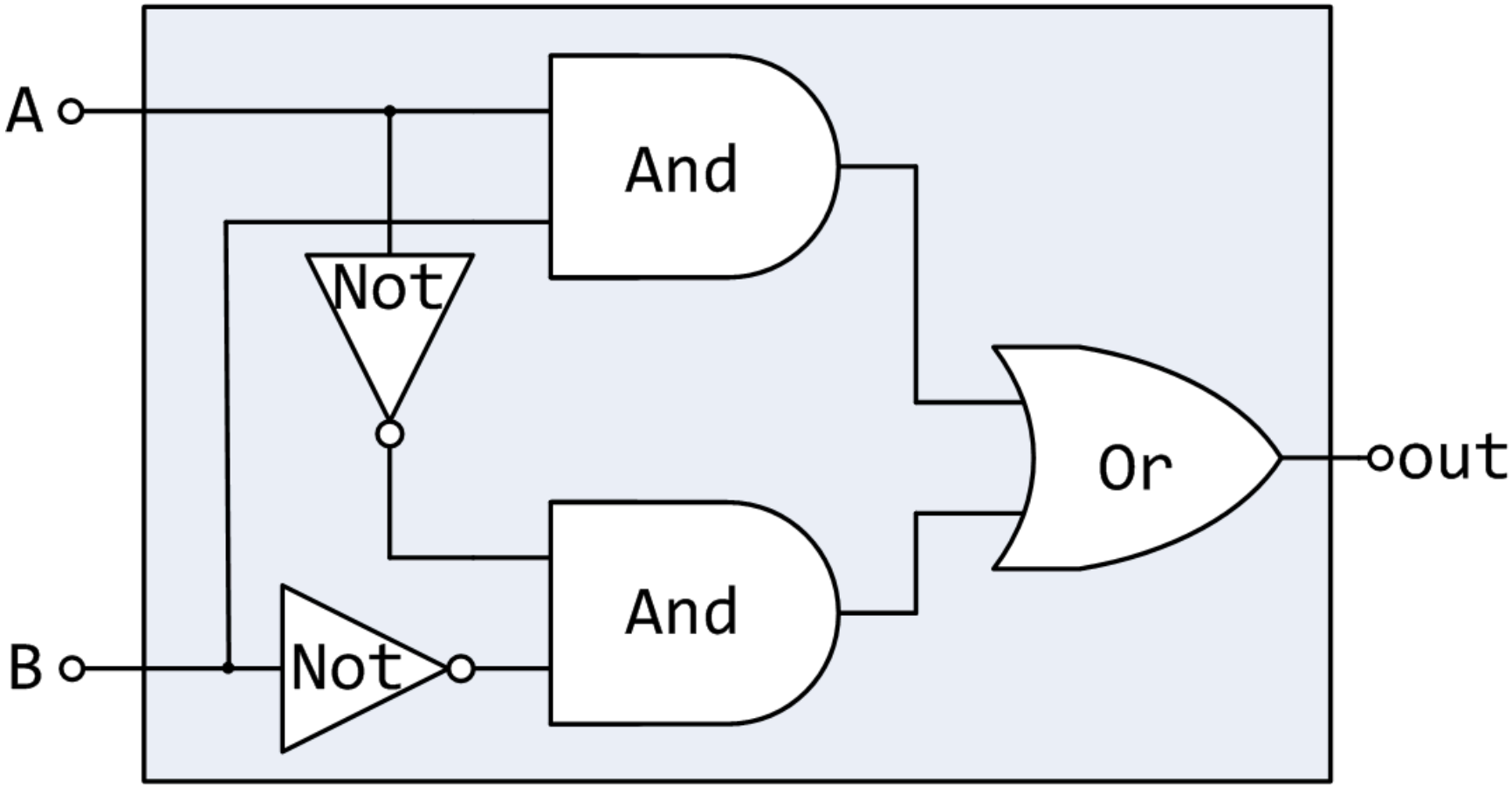




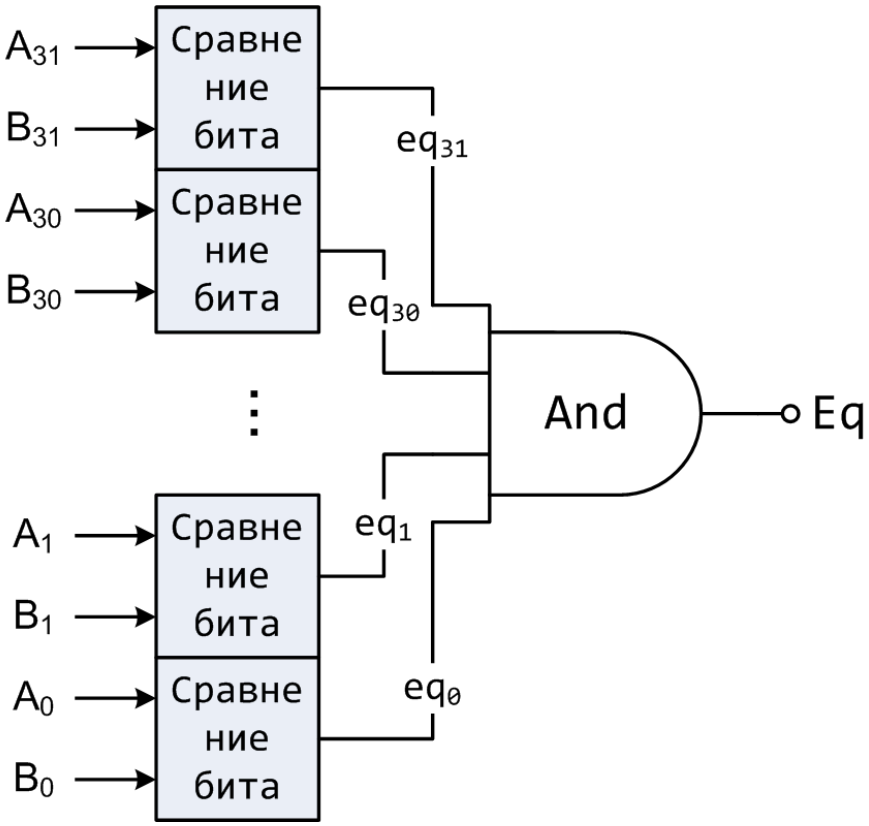


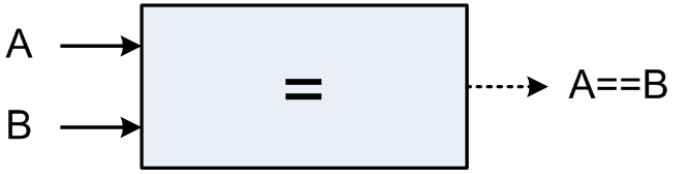


Сравнение битов



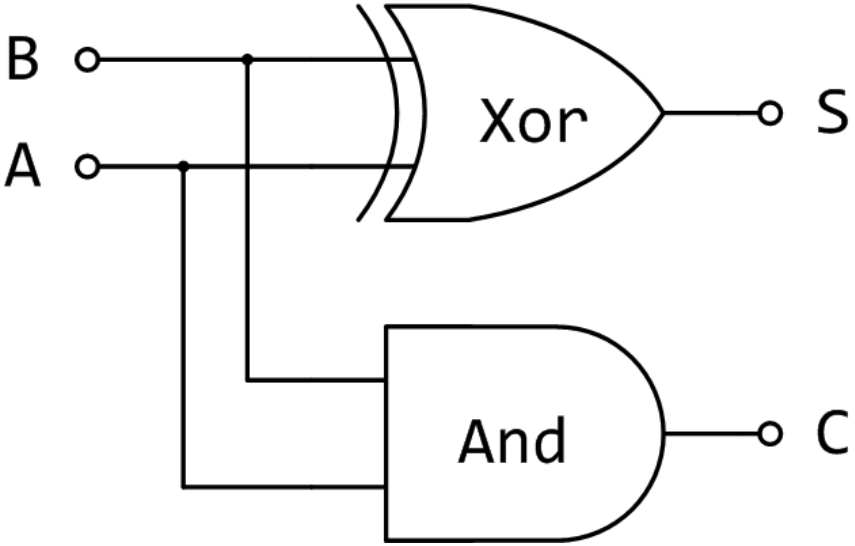
Сравнение слов



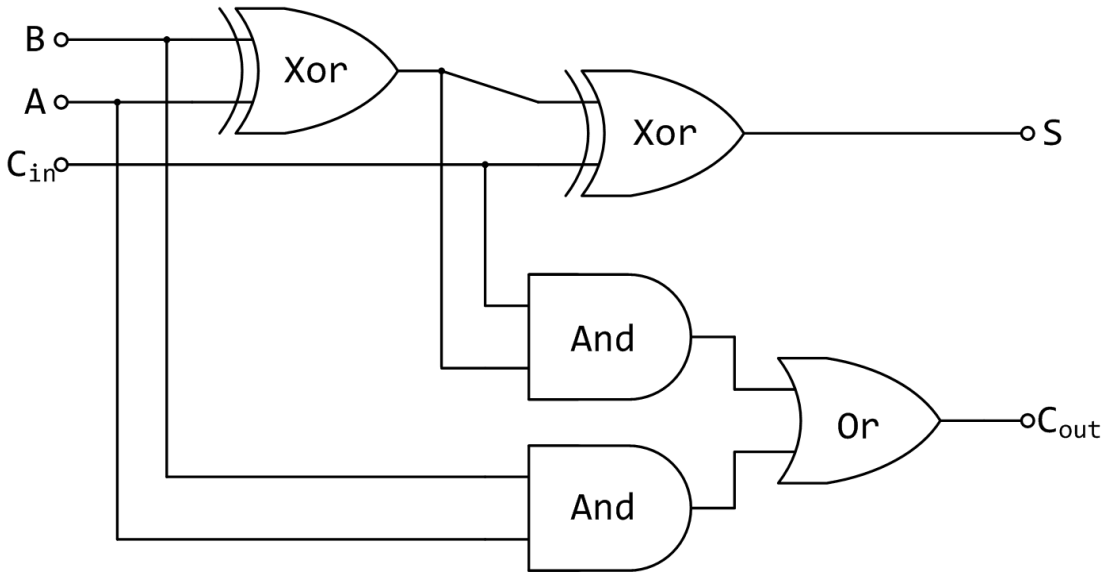


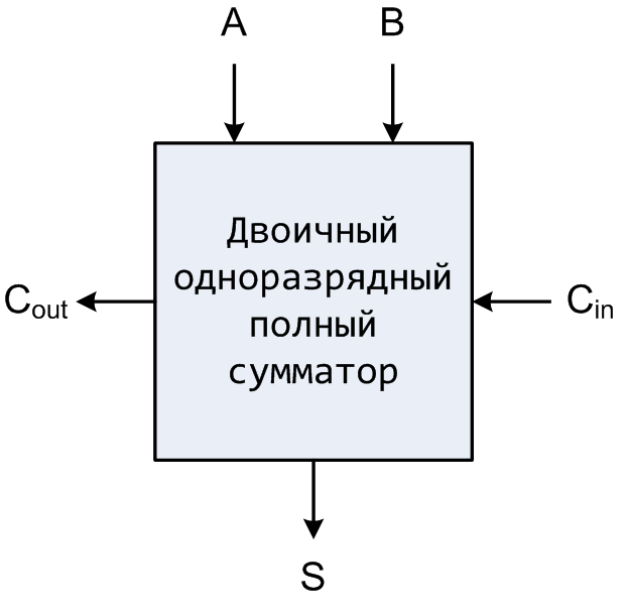
Полусумматор

- Суммирует биты A и B
- Результат – бит S
- Перенос – бит C

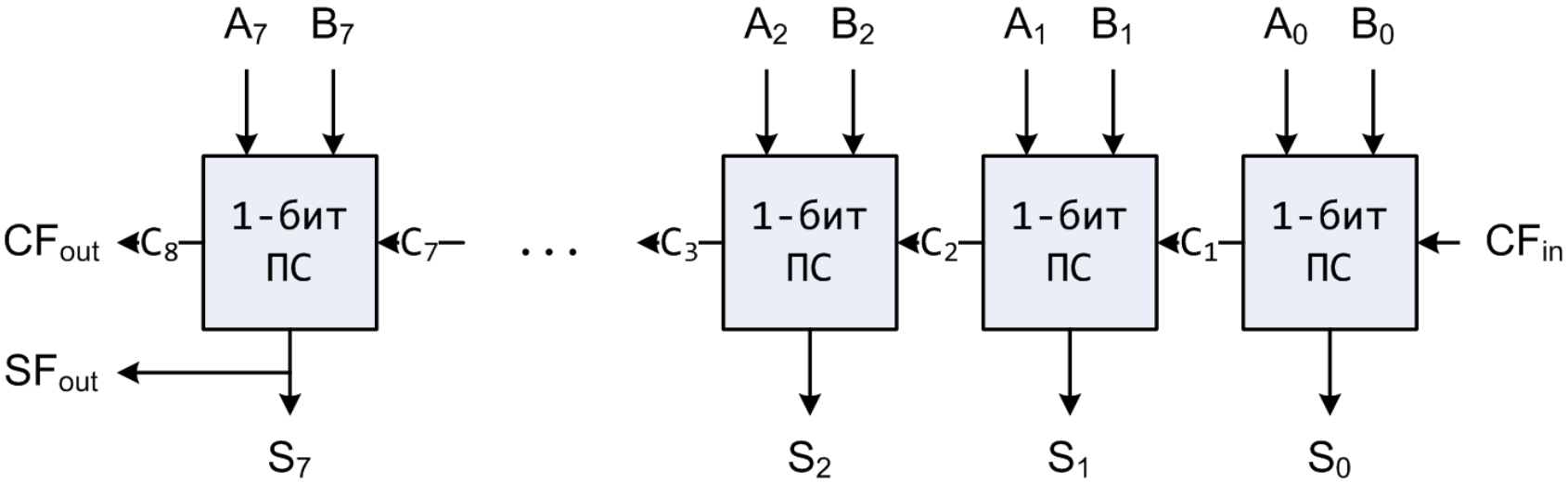


Полный двоичный сумматор

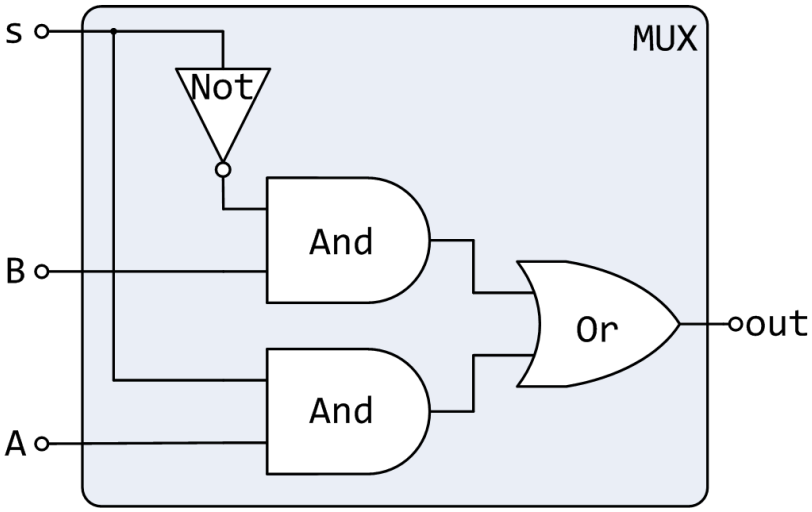


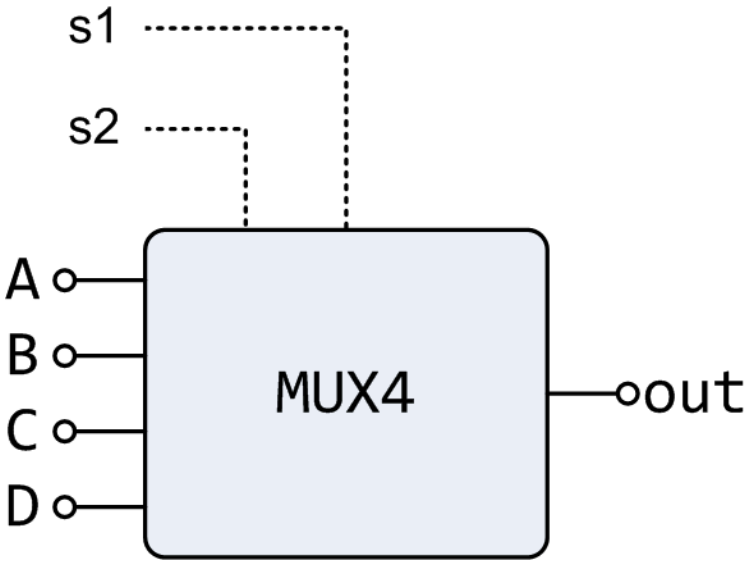


Суммирование слов

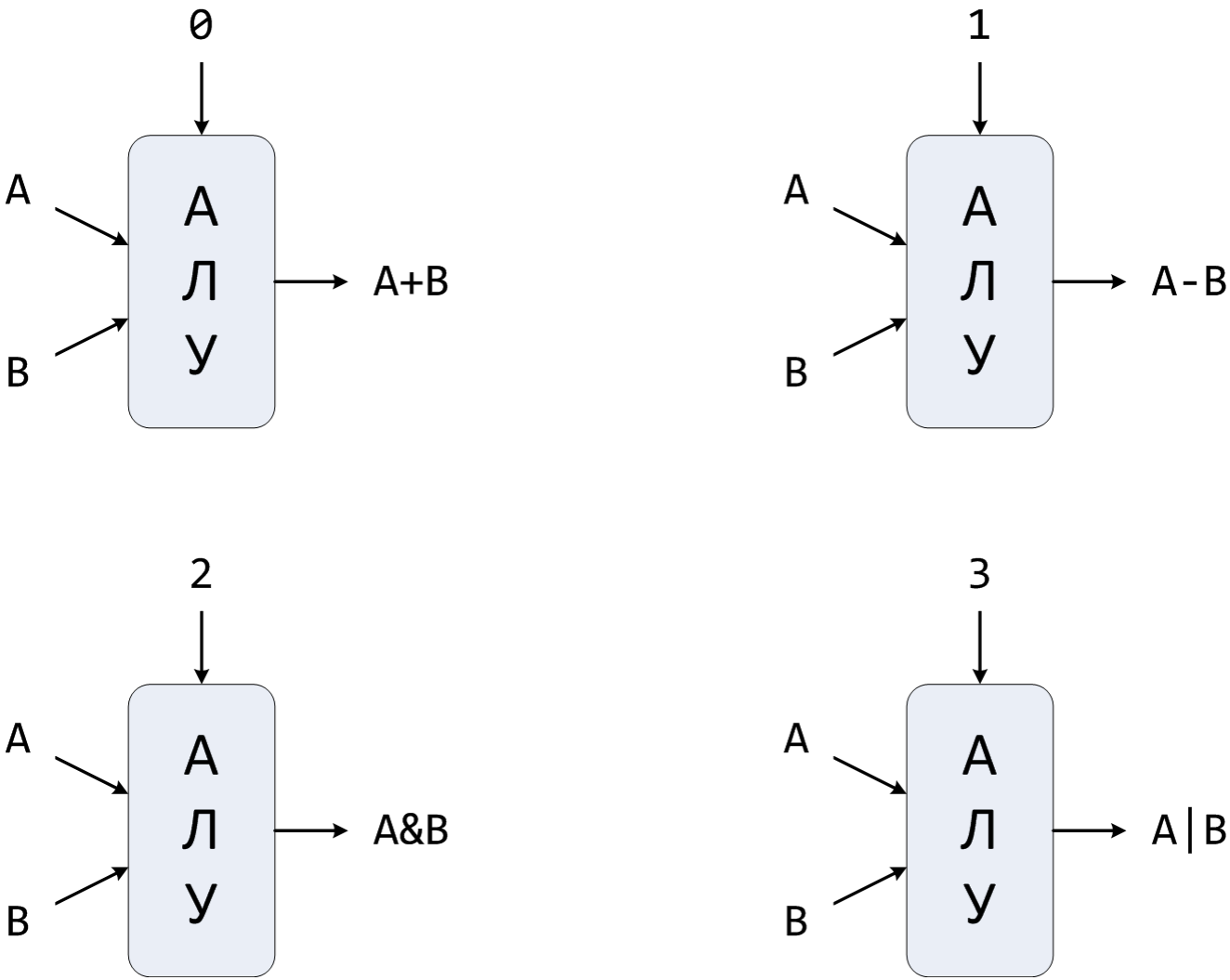


Мультиплексор

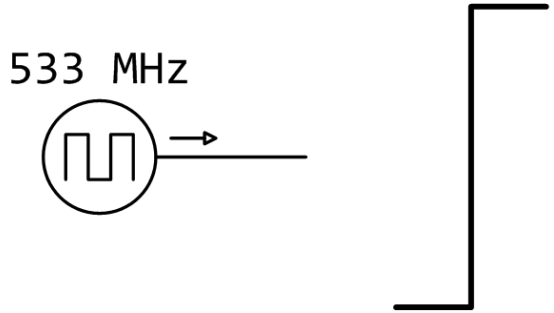




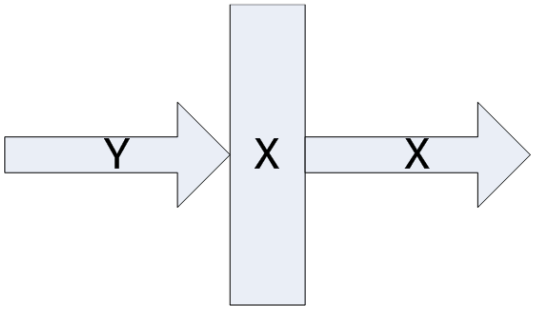
АЛУ



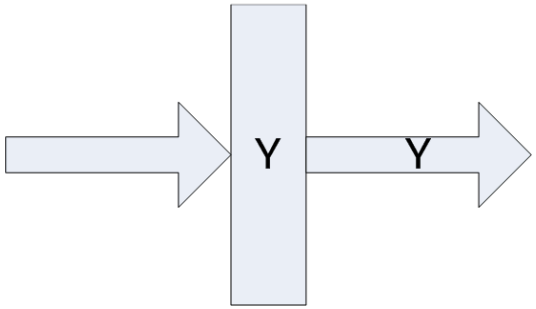
Регистр: сохранение 1 бита



Состояние регистра = X

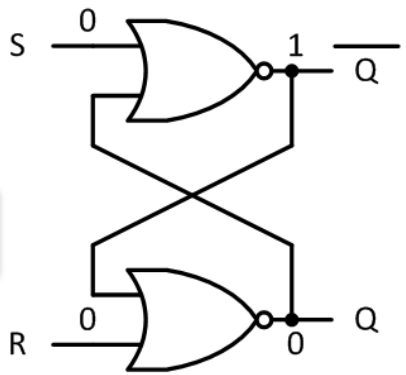


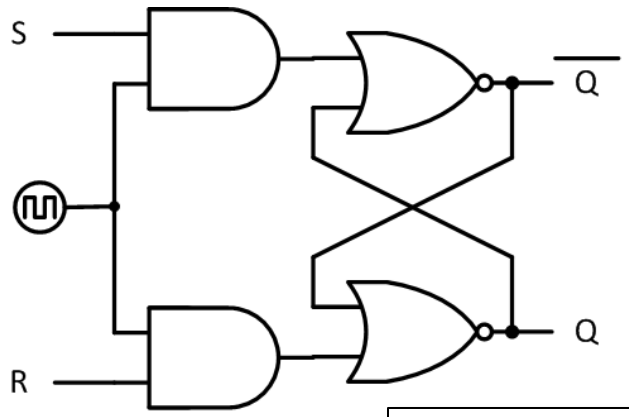
Состояние регистра = Y



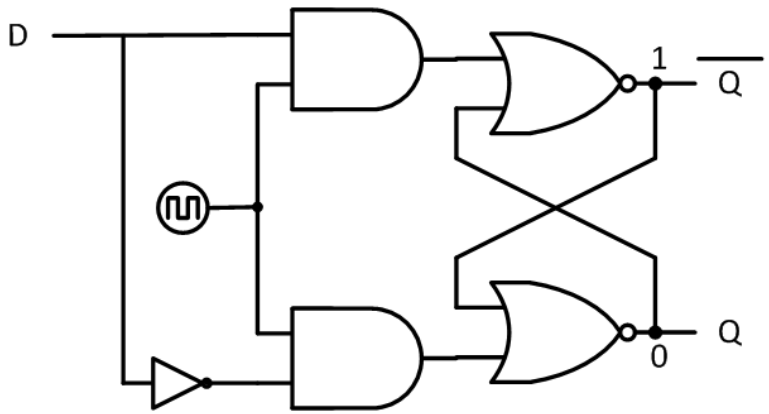
Статическая память

Защелка

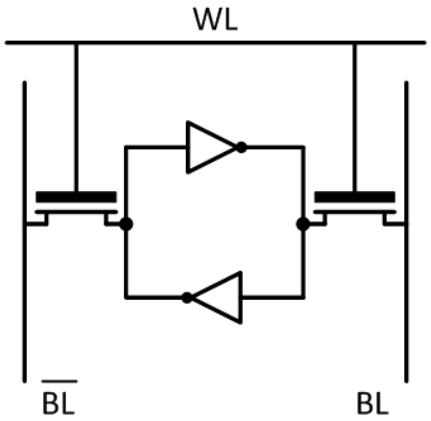




Синхронная SR-защелка



Синхронная D-защелка



Шеститранзисторная ячейка статической двоичной памяти (бит) SRAM

Динамическая память

- Один транзистор и один конденсатор
  - Занимает мало места
- Заряд утекает через транзистор
- Требуются циклы регенерации
  - чтение ячейки в качестве регенерации